

Quantum Pushdown Automata

Marats Golovkins

Institute of Mathematics and Computer Science, University of Latvia, Raiņa bulv.
29, Riga, Latvia*
marats@latnet.lv

Abstract. Quantum finite automata, as well as quantum pushdown automata (QPA) were first introduced by C. Moore, J. P. Crutchfield [MC 97]. In this paper we introduce the notion of QPA in a non-equivalent way, including unitarity criteria, by using the definition of quantum finite automata of [KW 97]. It is established that the unitarity criteria of QPA are not equivalent to the corresponding unitarity criteria of quantum Turing machines [BV 97]. We show that QPA can recognize every regular language. Finally we present some simple languages recognized by QPA, not recognizable by deterministic pushdown automata.

1 Introduction

Nobel prize winner physicist R. Feynman asked in 1982, what effects may have the principles of quantum mechanics on computation [Fe 82]. He gave arguments that it may require exponential time to simulate quantum mechanical processes on classical computers. This served as a basis to the opinion that quantum computers may have advantages versus classical ones. It was in 1985, when D. Deutsch introduced the notion of quantum Turing machine [De 85] and proved that quantum Turing machines compute the same recursive functions as classical deterministic Turing machines do. P. Shor discovered that by use of quantum algorithms it is possible to factorize large integers and compute discrete logarithms in a polynomial time [Sh 94], what resulted into additional interest in quantum computing and attempts to create quantum computers. First steps have been made to this direction, and first quantum computers which memory is limited by a few quantum bits have been constructed [KLMT 99]. To make quantum computers with larger memory feasible, one of the problems is to minimize error possibilities in quantum bits. Quantum error correction methods are developed [CRSS 98] which would enable quantum computers with larger quantum memory.

For the analysis of the current situation in quantum computation and information processing and main open issues one could see [Gr 99].

Quantum mechanics differs from the classical physics substantially. It is enough to mention *Heisenberg's uncertainty principle*, which states that it is

* Research partially supported by the Latvian Council of Science, grant 96-0282 and grant for PhD students; European Comission, contract IST-1999-11234; Swedish Institute, project ML2000

impossible to get information about different parameters of quantum particle simultaneously precisely. Another well known distinction is the impossibility to observe quantum object without changing it.

Fundamental concept of quantum information theory is *quantum bit*. Classical information theory is based on classical bit, which has two states 0 and 1. The next step is *probabilistic bit*, which can be 0 with probability α and 1 with probability β , where $\alpha + \beta = 1$. Quantum bit or *qbit* is similar to probabilistic bit with the difference that α and β are complex numbers with the property $|\alpha|^2 + |\beta|^2 = 1$. It is common to denote qbit as $\alpha|0\rangle + \beta|1\rangle$. As a result of *measurement*, we get 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.

Every computation done on qbits is accomplished by means of unitary operators. Informally, every unitary operator can be interpreted as a evolution in complex space. Therefore one of the basic properties of unitary operators is that every quantum computing process not disturbed by measurements is reversible. Unitarity is rather hard requirement which complicates programming of quantum devices. The following features of quantum computers are most important:

1. Information is represented by qbits.
2. Any step of computation can be represented as a unitary operation, therefore computation is reversible.
3. Quantum information cannot be copied.
4. Quantum parallelism; quantum computer can compute several paths simultaneously, however as a result of measurement it is possible to get the results of only one computation path.

Opposite to quantum Turing machines, quantum finite automata (QFA) represent the finite model of quantum computation. QFA were first introduced by [MC 97] (measure-once QFA), which were followed by a more elaborated model of [KW 97] (measure-many quantum finite automata). Since then QFA have been studied a lot, various properties of these automata are considered in [ABFK 99, AF 98, BP 99, Va 00]. Quantum finite one counter automata were introduced by [Kr 99].

The purpose of this paper is to introduce a quantum counterpart of pushdown automata, the next most important model after finite automata and Turing machines. The first definition of quantum pushdown automata was suggested by [MC 97], but here the authors actually deal with the so-called generalized quantum pushdown automata, which evolution does not have to be unitary. However a basic postulate of quantum mechanics imposes a strong constraint on any quantum machine model: it has to be unitary, otherwise it is questionable whether we can speak about *quantum* machine. That's why it was considered necessary to re-introduce quantum pushdown automata by giving a definition which would conform unitarity requirement. Such definition would enable us to study the properties of quantum pushdown automata.

The following notations will be used further in the paper:

z^* is the complex conjugate of a complex number z .

U^* is the Hermitian conjugate of a matrix U .

I is the identity matrix.

ε is empty word.

Definition 1.1. Matrix U is called unitary, if $UU^* = U^*U = I$.

If U is a finite matrix, then $UU^* = I$ iff $U^*U = I$. However this is not true for infinite matrices:

Example 1.1.

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \dots \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Here $U^*U = I$ but $UU^* \neq I$.

Lemma 1.1. If infinite matrices A, B, C have finite number of nonzero elements in each row and column, then their multiplication is associative: $(AB)C = A(BC)$.

Proof. The element of matrix $(AB)C$ in i -th row and j -th column is $k_{ij} = \sum_{s=1}^{\infty} \sum_{r=1}^{\infty} a_{ir} b_{rs} c_{sj}$. The element of matrix $A(BC)$ in the same row and column is $l_{ij} = \sum_{r=1}^{\infty} \sum_{s=1}^{\infty} a_{ir} b_{rs} c_{sj}$. As in the each row and column of matrices A, B, C there is a finite number of nonzero elements, it is also finite in the given series. Therefore the elements of the series can be rearranged, and $k_{ij} = l_{ij}$. \square

As noted further in the paper infinite matrices with finite number of nonzero elements in each row and column describe the work of pushdown automata. Further lemmas state some properties of such matrices.

Lemma 1.2. If $U^*U = I$, then the norm of any row in the matrix U does not exceed 1.

Proof. Let us consider the matrix $S = UU^*$. The element of this matrix $s_{ij} = \langle r_j | r_i \rangle$, where r_i is i -th row of the matrix U . Let us consider the matrix $T = S^2$. The diagonal element of this matrix is

$$t_{ii} = \sum_{k=1}^{\infty} s_{ik} s_{ki} = \sum_{k=1}^{\infty} \langle r_k | r_i \rangle \langle r_i | r_k \rangle = \sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2.$$

On the other hand, taking into account Lemma 1.1, we get that

$$T = S^2 = (UU^*)(UU^*) = U(U^*U)U^* = UU^* = S.$$

Therefore $t_{ii} = s_{ii} = \langle r_i | r_i \rangle$. It means that

$$\sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2 = \langle r_i | r_i \rangle. \quad (1)$$

This implies that every element of series (1) does not exceed $\langle r_i | r_i \rangle$. Hence $|\langle r_i | r_i \rangle|^2 = \langle r_i | r_i \rangle^2 \leq \langle r_i | r_i \rangle$. The last inequality implies that $0 \leq \langle r_i | r_i \rangle \leq 1$. Therefore $|r_i| \leq 1$. \square

Lemma 1.3. *Let us assume that $U^*U = I$. Then the rows of the matrix U are orthogonal iff every row of the matrix has norm 0 or 1.*

Proof. Let us assume that the rows of the matrix U are orthogonal. Let us consider equation (1) from the proof of Lemma 1.2, i.e., $\sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2 = \langle r_i | r_i \rangle$.

As the rows of the matrix U are orthogonal, $\sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2 = |\langle r_i | r_i \rangle|^2$. Hence $\langle r_i | r_i \rangle^2 = \langle r_i | r_i \rangle$, i.e., $\langle r_i | r_i \rangle = 0$ or $\langle r_i | r_i \rangle = 1$. Therefore $|r_i| = 0$ or $|r_i| = 1$.

Let us assume that every row of the matrix has norm 0 or 1. Then $\langle r_i | r_i \rangle^2 = \langle r_i | r_i \rangle$ and in compliance with the equation (1), $\sum_{k \in \mathbb{N}^+ \setminus \{i\}} |\langle r_k | r_i \rangle|^2 = 0$. This implies that $\forall k \neq i \quad |\langle r_k | r_i \rangle| = 0$. Hence the rows of the matrix are orthogonal. \square

Lemma 1.4. *The matrix U is unitary iff $U^*U = I$ and its rows are normalized.*

Proof. Let us assume that the matrix U is unitary. Then in compliance with Definition 1.1, $U^*U = I$ and $UU^* = I$, i.e., the rows of the matrix are orthonormal.

Let us assume that $U^*U = I$ and the rows of the matrix are normalized. Then in compliance with Lemma 1.3 the rows of the matrix are orthogonal. Hence $UU^* = I$ and the matrix is unitary. \square

This result is very similar to Lemma 1 of [DS 96].

2 Quantum pushdown automata

Definition 2.1. *A quantum pushdown automaton (QPA)*

$A = (Q, \Sigma, T, q_0, Q_a, Q_r, \delta)$ is specified by a finite set of states Q , a finite input alphabet Σ and a stack alphabet T , an initial state $q_0 \in Q$, sets $Q_a \subset Q$, $Q_r \subset Q$ of accepting and rejecting states, respectively, with $Q_a \cap Q_r = \emptyset$, and a transition function

$$\delta : Q \times \Gamma \times \Delta \times Q \times \{\downarrow, \rightarrow\} \times \Delta^* \longrightarrow \mathbb{C}_{[0,1]},$$

where $\Gamma = \Sigma \cup \{\#, \$\}$ is the input tape alphabet of A and $\#, \$$ are end-markers not in Σ , $\Delta = T \cup \{Z_0\}$ is the working stack alphabet of A and $Z_0 \notin T$ is

the stack base symbol; $\{\downarrow, \rightarrow\}$ is the set of directions of input tape head. The automaton must satisfy conditions of well-formedness, which will be expressed below. Furthermore, the transition function is restricted to a following requirement:

If $\delta(q, \alpha, \beta, q', d, \omega) \neq 0$, then

1. $|\omega| \leq 2$;
2. if $|\omega| = 2$, then $\omega_1 = \beta$;
3. if $\beta = Z_0$, then $\omega \in Z_0 T^*$;
4. if $\beta \neq Z_0$, then $\omega \in T^*$.

Definition 2.1 utilizes that of classical pushdown automata from [Gu 89].

Well-formedness conditions 2.1.

1. Local probability condition.

$$\begin{aligned} \forall (q_1, \sigma_1, \tau_1) \in Q \times \Gamma \times \Delta \\ \sum_{(q, d, \omega) \in Q \times \{\downarrow, \rightarrow\} \times \Delta^*} |\delta(q_1, \sigma_1, \tau_1, q, d, \omega)|^2 = 1. \end{aligned} \quad (2)$$

2. Orthogonality of column vectors condition.

$$\begin{aligned} \text{For all triples } (q_1, \sigma_1, \tau_1) \neq (q_2, \sigma_1, \tau_2) \text{ in } Q \times \Gamma \times \Delta \\ \sum_{(q, d, \omega) \in Q \times \{\downarrow, \rightarrow\} \times \Delta^*} \delta^*(q_1, \sigma_1, \tau_1, q, d, \omega) \delta(q_2, \sigma_1, \tau_2, q, d, \omega) = 0. \end{aligned} \quad (3)$$

3. Row vectors norm condition.

$$\begin{aligned} \forall (q_1, \sigma_1, \sigma_2, \tau_1, \tau_2) \in Q \times \Gamma^2 \times \Delta^2 \\ \sum_{(q, \tau, \omega) \in Q \times \Delta \times \{\varepsilon, \tau_2, \tau_1 \tau_2\}} |\delta(q, \sigma_1, \tau, q_1, \rightarrow, \omega)|^2 + |\delta(q, \sigma_2, \tau, q_1, \downarrow, \omega)|^2 = 1. \end{aligned} \quad (4)$$

4. Separability condition I.

$$\begin{aligned} \forall (q_1, \sigma_1, \tau_1), (q_2, \sigma_1, \tau_2) \in Q \times \Gamma \times \Delta, \forall \tau_3 \in \Delta \\ a) \sum_{(q, d, \tau) \in Q \times \{\downarrow, \rightarrow\} \times \Delta} \delta^*(q_1, \sigma_1, \tau_1, q, d, \tau) \delta(q_2, \sigma_1, \tau_2, q, d, \tau_3 \tau) + \\ + \sum_{(q, d) \in Q \times \{\downarrow, \rightarrow\}} \delta^*(q_1, \sigma_1, \tau_1, q, d, \varepsilon) \delta(q_2, \sigma_1, \tau_2, q, d, \tau_3) = 0; \end{aligned} \quad (5)$$

$$b) \sum_{(q, d) \in Q \times \{\downarrow, \rightarrow\}} \delta^*(q_1, \sigma_1, \tau_1, q, d, \varepsilon) \delta(q_2, \sigma_1, \tau_2, q, d, \tau_2 \tau_3) = 0. \quad (6)$$

5. Separability condition II.

$$\begin{aligned} \forall (q_1, \sigma_1, \tau_1), (q_2, \sigma_2, \tau_2) \in Q \times \Gamma \times \Delta \\ \sum_{(q, \omega) \in Q \times \Delta^*} \delta^*(q_1, \sigma_1, \tau_1, q, \downarrow, \omega) \delta(q_2, \sigma_2, \tau_2, q, \rightarrow, \omega) = 0. \end{aligned} \quad (7)$$

6. Separability condition III.

$$\begin{aligned} & \forall (q_1, \sigma_1, \tau_1), (q_2, \sigma_2, \tau_2) \in Q \times \Gamma \times \Delta, \forall \tau_3 \in \Delta, \forall d_1, d_2 \in \{\downarrow, \rightarrow\}, d_1 \neq d_2 \\ a) & \sum_{(q, \tau) \in Q \times \Delta} \delta^*(q_1, \sigma_1, \tau_1, q, d_1, \tau) \delta(q_2, \sigma_2, \tau_2, q, d_2, \tau_3 \tau) + \\ & + \sum_{q \in Q} \delta^*(q_1, \sigma_1, \tau_1, q, d_1, \varepsilon) \delta(q_2, \sigma_2, \tau_2, q, d_2, \tau_3) = 0; \end{aligned} \quad (8)$$

$$b) \sum_{q \in Q} \delta^*(q_1, \sigma_1, \tau_1, q, d_1, \varepsilon) \delta(q_2, \sigma_2, \tau_2, q, d_2, \tau_2 \tau_3) = 0. \quad (9)$$

Let us assume that an automaton is in a state q , its input tape head is above a symbol α and the stack head is above a symbol β . Then the automaton undertakes following actions with an amplitude $\delta(q, \alpha, \beta, q', d, \omega)$:

1. goes into the state q' ;
2. if $d = \rightarrow$, moves the input tape head one cell forward;
3. takes out of the stack the symbol β (deletes it and moves the stack head one cell backwards);
4. starting with the first empty cell, puts into the stack the string ω , moving the stack head $|\omega|$ cells forward.

Definition 2.2. *The configuration of a pushdown automaton is a pair $|c\rangle = |\nu_i q_j \nu_k, \omega_l\rangle$, where the automaton is in a state $q_j \in Q$, $\nu_i \nu_k \in \# \Sigma^* \$$ is a finite word on the input tape, $\omega_l \in Z_0 T^*$ is a finite word on the stack tape, the input tape head is above the first symbol of the word ν_k and the stack head is above the last symbol of the word ω_l .*

We shall denote by C the set of all configurations of a pushdown automaton. The set C is countably infinite. Every configuration $|c\rangle$ denotes a basis vector in the space $H_A = l_2(C)$. Therefore a global state of A in the space H_A has a form $|\psi\rangle = \sum_{c \in C} \alpha_c |c\rangle$, where $\sum_{c \in C} |\alpha_c|^2 = 1$ and $\alpha_c \in \mathbb{C}$ denotes the amplitude of a configuration $|c\rangle$. If an automaton is in its global state (superposition) $|\psi\rangle$, then its further step is equivalent to the application of a linear operator (evolution) U_A over the space H_A .

Definition 2.3. *A linear operator U_A is defined as follows:*

$$U_A |\psi\rangle = \sum_{c \in C} \alpha_c U_A |c\rangle.$$

If a configuration $c = |\nu_i q_j \sigma \nu_k, \omega_l \tau\rangle$, then

$$U_A |c\rangle = \sum_{(q, d, \omega) \in Q \times \{\downarrow, \rightarrow\} \times \Delta^*} \delta(q_j, \sigma, \tau, q, d, \omega) |f(|c\rangle, d, q), \omega_l \omega\rangle,$$

where

$$f(|\nu_i q_j \sigma \nu_k, \omega_l \tau\rangle, d, q) = \begin{cases} \nu_i q \sigma \nu_k, & \text{if } d = \downarrow, \\ \nu_i \sigma q \nu_k, & \text{if } d = \rightarrow. \end{cases}$$

Remark 2.1. Although a QPA evolution operator matrix is infinite, it has a finite number of nonzero elements in each row and column, as it is possible to reach only a finite number of other configurations from a given configuration within one step, all the same, within one step the given configuration is reachable only from a finite number of different configurations.

Lemma 2.1. *The columns system of a QPA evolution matrix is normalized iff the condition (2), i.e., local probability condition, is satisfied.*

Lemma 2.2. *The columns system of a QPA evolution matrix is orthogonal iff the conditions (3,5,6,7,8,9), i.e., orthogonality of column vectors and separability conditions, are satisfied.*

Lemma 2.3. *The rows system of a QPA evolution matrix is normalized iff the condition (4), i.e., row vectors norm condition, is satisfied.*

Theorem 2.1. *Well-formedness conditions 2.1 are satisfied iff the evolution operator U_A is unitary.*

Proof. Lemmas 2.1, 2.2, 2.3 imply that Well-formedness conditions 2.1 are satisfied iff the columns of the evolution matrix are orthonormal and rows are normalized. In compliance with Lemma 1.4, columns are orthonormal and rows are normalized iff the matrix is unitary. \square

Remark 2.2. Well-formedness conditions 2.1 contain the requirement that rows system has to be normalized, which is not necessary in the case of quantum Turing machine [BV 97]. Here is taken into account the fact that the evolution of QPA can violate the unitarity requirement if the row vectors norm condition is omitted.

Example 2.1. A QPA, whose evolution matrix columns are orthonormal, however the evolution is not unitary.

$$\begin{aligned} Q &= \{q\}, \quad \Sigma = \{1\}, \quad T = \{1\}. \\ \delta(q, \#, Z_0, q, \rightarrow, Z_0 1) &= 1, & \delta(q, \#, 1, q, \rightarrow, 11) &= 1, \\ \delta(q, 1, Z_0, q, \rightarrow, Z_0 1) &= 1, & \delta(q, 1, 1, q, \rightarrow, 11) &= 1, \\ \delta(q, \$, Z_0, q, \rightarrow, Z_0 1) &= 1, & \delta(q, \$, 1, q, \rightarrow, 11) &= 1, \end{aligned}$$

other values of arguments yield $\delta = 0$.

By Well-formedness conditions 2.1, the columns of the evolution matrix are orthonormal, but the matrix is not unitary, because the norm of the rows specified by the configurations $|\omega, Z_0\rangle$ is 0.

Even in a case of trivial QPA, it is a cumbersome task to check all the conditions of well-formedness 2.1. It is possible to relax the conditions slightly by introducing a notion of *simplified* QPA.

Definition 2.4. We shall say that a QPA is simplified, if there exists a function $D : Q \longrightarrow \{\downarrow, \rightarrow\}$, and $\delta(q_1, \sigma, \tau, q, d, \omega) = 0$, if $D(q) \neq d$. Therefore the transition function of a simplified QPA is

$$\varphi(q_1, \sigma, \tau, q, \omega) = \delta(q_1, \sigma, \tau, q, D(q), \omega).$$

Taking into account Definition 2.4, following well-formedness conditions correspond to simplified QPA:

Well-formedness conditions 2.2.

1. Local probability condition.

$$\begin{aligned} \forall (q_1, \sigma_1, \tau_1) \in Q \times \Gamma \times \Delta \\ \sum_{(q, \omega) \in Q \times \Delta^*} |\varphi(q_1, \sigma_1, \tau_1, q, \omega)|^2 = 1. \end{aligned} \quad (10)$$

2. Orthogonality of column vectors condition.

$$\begin{aligned} \text{For all triples } (q_1, \sigma_1, \tau_1) \neq (q_2, \sigma_1, \tau_2) \text{ in } Q \times \Gamma \times \Delta \\ \sum_{(q, \omega) \in Q \times \Delta^*} \varphi^*(q_1, \sigma_1, \tau_1, q, \omega) \varphi(q_2, \sigma_1, \tau_2, q, \omega) = 0. \end{aligned} \quad (11)$$

3. Row vectors norm condition.

$$\begin{aligned} \forall (q_1, \sigma_1, \tau_1, \tau_2) \in Q \times \Gamma \times \Delta^2 \\ \sum_{(q, \tau, \omega) \in Q \times \Delta \times \{\varepsilon, \tau_2, \tau_1 \tau_2\}} |\varphi(q, \sigma_1, \tau, q_1, \omega)|^2 = 1. \end{aligned} \quad (12)$$

4. Separability condition.

$$\begin{aligned} \forall (q_1, \sigma_1, \tau_1), (q_2, \sigma_1, \tau_2) \in Q \times \Gamma \times \Delta, \forall \tau_3 \in \Delta \\ a) \sum_{(q, \tau) \in Q \times \Delta} \varphi^*(q_1, \sigma_1, \tau_1, q, \tau) \varphi(q_2, \sigma_1, \tau_2, q, \tau_3 \tau) + \\ + \sum_{q \in Q} \varphi^*(q_1, \sigma_1, \tau_1, q, \varepsilon) \varphi(q_2, \sigma_1, \tau_2, q, \tau_3) = 0; \end{aligned} \quad (13)$$

$$b) \sum_{q \in Q} \varphi^*(q_1, \sigma_1, \tau_1, q, \varepsilon) \varphi(q_2, \sigma_1, \tau_2, q, \tau_2 \tau_3) = 0. \quad (14)$$

Theorem 2.2. The evolution of a simplified QPA is unitary iff Well-formedness conditions 2.2 are satisfied.

Proof. By Theorem 2.1 and Definition 2.4. □

3 Language recognition

Language recognition for QPA is defined as follows. For a QPA

$A = (Q, \Sigma, T, q_0, Q_a, Q_r, \delta)$ we define $C_a = \{|\nu_i q \nu_k, \omega_l\rangle \in C \mid q \in Q_a\}$, $C_r = \{|\nu_i q \nu_k, \omega_l\rangle \in C \mid q \in Q_r\}$, $C_n = C \setminus (C_a \cup C_r)$. E_a, E_r, E_n are subspaces of H_A spanned by C_a, C_r, C_n respectively. We use the observable \mathcal{O} that corresponds to the orthogonal decomposition $H_A = E_a \oplus E_r \oplus E_n$. The outcome of each observation is either “accept” or “reject” or “non-halting”.

The language recognition is now defined as follows: For an $x \in \Sigma^*$ we consider as an input $\#x\$$, and assume that the computation starts with A being in the configuration $|q_0 \#x\$, Z_0\rangle$. Each computation step consists of two parts. At first the linear operator U_A is applied to the current global state and then the resulting superposition is observed using the observable \mathcal{O} as defined above. If the global state before the observation is $\sum_{c \in C} \alpha_c |c\rangle$, then the probability that the resulting superposition is projected into the subspace E_i , $i \in \{a, r, n\}$, is $\sum_{c \in C_i} |\alpha_c|^2$. The computation continues until the result of an observation is “accept” or “reject”.

Definition 3.1. *We shall say that an automaton is a deterministic reversible pushdown automaton (RPA), if it is a simplified QPA with $\varphi(q_1, \sigma, \tau, q, \omega) \in \{0, 1\}$ and there exists a function $f : Q \times \Gamma \times \Delta \longrightarrow Q \times \Delta^*$, such that $f(q_1, \sigma, \tau) = (q, \omega)$ if and only if $\varphi(q_1, \sigma, \tau, q, \omega) = 1$.*

We can regard f as a transition function of a RPA. Needless to say, if any language is recognized by a RPA, it is recognized with probability equal to 1. Note that the local probability condition (10) is satisfied automatically for RPA.

Theorem 3.1. *Every regular language is recognizable by some QPA.*

Proof. It is sufficient to prove that any deterministic finite automaton (DFA) can be simulated by RPA. Let us consider a DFA with n states $A_{DFA} = (Q_{DFA}, \Sigma, q_0, Q_F, \delta)$, where $\delta : Q_{DFA} \times \Sigma \longrightarrow Q_{DFA}$.

To simulate A_{DFA} we shall construct a RPA $A_{RPA} = (Q, \Sigma, T, q_0, Q_a, Q_r, \varphi)$ with the number of states $2n$.

The set of states is $Q = Q_{DFA} \cup Q'_{DFA}$, where Q'_{DFA} are the newly introduced states, which are linked to Q_{DFA} by a one-to-one relation $\{(q_i, q'_i) \in Q_{DFA} \times Q'_{DFA}\}$. Thus Q_F has one-to-one relation to $Q'_F \subset Q'_{DFA}$.

The stack alphabet is $T = Ind(Q_{DFA})$, where $\forall i \text{ } Ind(q_i) = i$; the set of accepting states is $Q_a = Q'_F$ and the set of rejecting states is $Q_r = Q'_{DFA} \setminus Q'_F$. As for the function D , $D(Q_{DFA}) = \{\rightarrow\}$ and $D(Q'_{DFA}) = \{\downarrow\}$.

We shall define sets R and \overline{R} as follows:

$$R = \{(q'_j, \sigma, i) \in Q'_{DFA} \times \Sigma \times T \mid \delta(q_i, \sigma) = q_j\};$$

$$\overline{R} = \{(q'_j, \sigma, i) \in Q'_{DFA} \times \Sigma \times T \mid \delta(q_i, \sigma) \neq q_j\}.$$

The construction of the transition function f is performed by the following rules:

1. $\forall (q_i, \sigma, \tau) \in Q_{DFA} \times \Sigma \times \Delta \quad f(q_i, \sigma, \tau) = (\delta(q_i, \sigma), \tau i);$
2. $\forall (q'_j, \sigma, i) \in R \quad f(q'_j, \sigma, i) = (q'_i, \varepsilon);$
3. $\forall (q'_j, \sigma, i) \in \overline{R} \quad f(q'_j, \sigma, i) = (q_j, i);$
4. $\forall (q'_j, \sigma) \in Q'_{DFA} \times \Sigma \quad f(q'_j, \sigma, Z) = (q_j, Z);$
5. $\forall (q, \tau) \in Q \times \Delta \quad f(q, \#, \tau) = (q, \tau);$
6. $\forall (q_i, \tau) \in Q_{DFA} \times \Delta \quad f(q_i, \$, \tau) = (q'_i, \tau);$
7. $\forall (q'_i, \tau) \in Q'_{DFA} \times \Delta \quad f(q'_i, \$, \tau) = (q_i, \tau).$

Thus we have defined f for all the possible arguments. Our automaton simulates the DFA. Note that the automaton may reach a state in Q'_{DFA} only by reading the end-marking symbol $\$$ on the input tape. As soon as A_{RPA} reaches the end-marking symbol $\$$, it goes to an accepting state, if its current state is in Q_F , and goes to a rejecting state otherwise.

The construction is performed in a way so that A_{RPA} satisfies Well-formedness conditions 2.2.

As we know, RPA automatically satisfies the local probability condition (10).

Let us prove, that the automaton satisfies the orthogonality condition (11).

For RPA, the condition (11) is equivalent to the requirement that for all triples $(q_1, \sigma_1, \tau_1) \neq (q_2, \sigma_1, \tau_2) \quad f(q_1, \sigma_1, \tau_1) \neq f(q_2, \sigma_1, \tau_2)$.

If $q_1, q_2 \in Q_{DFA}$, $f(q_1, \sigma_1, \tau_1) \neq f(q_2, \sigma_1, \tau_2)$ by rule 1.

Let us consider the case when $(q_1, \sigma_1, \tau_1), (q_2, \sigma_1, \tau_2) \in R$. We shall denote q_1, q_2 as q'_i, q'_j respectively. Let us assume from the contrary that $f(q'_i, \sigma_1, \tau_1) = f(q'_j, \sigma_1, \tau_2)$. By rule 2, $(q'_{\tau_1}, \varepsilon) = (q'_{\tau_2}, \varepsilon)$. Hence $\tau_1 = \tau_2$. By the definition of R , $\delta(q_{\tau_1}, \sigma_1) = q_i$ and $\delta(q_{\tau_2}, \sigma_1) = q_j$. Since $\tau_1 = \tau_2$, $q_i = q_j$. Therefore $q'_i = q'_j$, i.e., $q_1 = q_2$. We have come to a contradiction with the fact that $(q_1, \sigma_1, \tau_1) \neq (q_2, \sigma_1, \tau_2)$.

If $(q_1, \sigma_1, \tau_1), (q_2, \sigma_1, \tau_2) \in \overline{R}$, $f(q_1, \sigma_1, \tau_1) \neq f(q_2, \sigma_1, \tau_2)$ by rule 3.

If $q_1 \in Q_{DFA}, q_2 \in Q'_{DFA}$ then $f(q_1, \sigma_1, \tau_1) \neq f(q_2, \sigma_1, \tau_2)$ by rules 1, 2, 3.

In case τ_1 or τ_2 is Z , or $\sigma_1 \in \{\#, \$\}$, proof is straightforward.

The compliance with row vectors norm condition (12) and separability conditions (13) and (14) is proved in the same way. \square

Example 3.1. First, let us consider a language $L_1 = (0, 1)^*1$, for which we know that it is not recognizable by QFA [KW 97].

Language L_1 is recognizable by a RPA. Let us consider a deterministic finite automaton with two states q_0, q_1 and the following transitions: $\delta(q_0, 0) = q_0$, $\delta(q_0, 1) = q_1$, $\delta(q_1, 0) = q_0$, $\delta(q_1, 1) = q_1$.

It is possible to transform this automaton to a RPA, which satisfies the corresponding well-formedness conditions:

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $Q_a = \{q_5\}$, $Q_r = \{q_4\}$, $\Sigma = \{0, 1\}$, $T = \{0, 1\}$.

The states q_0, q_1 have the same semantics as in the deterministic prototype, the only difference is in case input tape symbols 0 or 1 is read, when each transition starting in the state q_0 , automaton pushes 0 into stack, whereas in the state q_1 pushes 1. After reaching the endmarking symbol $\$$, depending on its current state, the automaton goes to the state q_5 or q_6 .

Finally, we have to add two more states q_2, q_3 to our RPA, to ensure its unitarity. Values of the transition function follow:

$$\begin{aligned}
& \forall \tau \in \Delta \quad \forall q \in Q \quad \forall \sigma \in \Sigma; \\
& \varphi(q, \#, \tau, q, \tau) = 1, \\
& \varphi(q_0, 0, \tau, q_0, \tau 0) = 1, \quad \varphi(q_1, 0, \tau, q_0, \tau 1) = 1, \quad D(q_0) = \rightarrow, \\
& \varphi(q_0, 1, \tau, q_1, \tau 0) = 1, \quad \varphi(q_1, 1, \tau, q_1, \tau 1) = 1, \quad D(q_1) = \rightarrow, \\
& \varphi(q_0, \$, \tau, q_4, \tau) = 1, \quad \varphi(q_1, \$, \tau, q_5, \tau) = 1, \quad D(q_2) = \downarrow, \\
& \varphi(q_2, 1, \tau, q_0, \tau) = 1, \quad \varphi(q_3, 0, \tau, q_1, \tau) = 1, \quad D(q_3) = \downarrow, \\
& \varphi(q_2, \$, \tau, q_2, \tau) = 1, \quad \varphi(q_3, \$, \tau, q_3, \tau) = 1, \quad D(q_4) = \downarrow, \\
& \varphi(q_4, \sigma, \tau, q_4, \tau) = 1, \quad \varphi(q_5, \sigma, \tau, q_5, \tau) = 1, \quad D(q_5) = \downarrow, \\
& \varphi(q_2, 0, Z, q_0, Z) = 1, \quad \varphi(q_3, 1, Z, q_1, Z) = 1, \quad D(q_6) = \downarrow, \\
& \varphi(q_2, 0, 0, q_2, \varepsilon) = 1, \quad \varphi(q_2, 0, 1, q_3, \varepsilon) = 1, \\
& \varphi(q_3, 1, 0, q_2, \varepsilon) = 1, \quad \varphi(q_3, 1, 1, q_3, \varepsilon) = 1, \\
& \varphi(q_4, \$, \tau, q_0, \tau) = 1, \quad \varphi(q_5, \$, \tau, q_1, \tau) = 1,
\end{aligned}$$

other values of arguments yield $\delta = 0$.

Let us note that states q_2, q_3 are not reachable from the initial state q_0 , however they are necessary to make the automaton unitary. \square

Let us consider a language which is not regular, namely,

$$L_2 = \{\omega \in (a, b)^* \mid |\omega|_a = |\omega|_b\},$$

where $|\omega|_i$ denotes the number of occurrences of the symbol i in the word ω .

Lemma 3.1. *Language L_2 is recognizable by a RPA.*

Proof. Our RPA has four states q_0, q_1, q_2, q_3 , where q_2 is an accepting state, whereas q_3 - rejecting one. Stack alphabet T consists of two symbols 1, 2. Stack filled with 1's means that the processed part of the word ω has more occurrences of a's than b's, whereas 2's means that there are more b's than a's. Furthermore, length of the stack word is equal to the difference of number of a's and b's. Empty stack denotes that the number of a's and b's is equal.

Values of the transition function follow:

$$\begin{aligned}
& \forall q \in Q \quad \forall \tau \in \Delta; \\
& \varphi(q, \#, \tau, q, \tau) = 1, \quad \varphi(q_0, a, Z, q_0, Z1) = 1, \quad D(q_0) = \rightarrow, \\
& \varphi(q_0, b, Z, q_0, Z2) = 1, \quad \varphi(q_0, \$, Z, q_2, Z1) = 1, \quad D(q_1) = \downarrow, \\
& \varphi(q_0, a, 1, q_0, 11) = 1, \quad \varphi(q_0, b, 1, q_1, \varepsilon) = 1, \quad D(q_2) = \downarrow, \\
& \varphi(q_0, \$, 1, q_3, 1) = 1, \quad \varphi(q_0, a, 2, q_1, \varepsilon) = 1, \quad D(q_3) = \downarrow, \\
& \varphi(q_0, b, 2, q_0, 22) = 1, \quad \varphi(q_0, \$, 2, q_3, 2) = 1, \quad \varphi(q_1, a, Z, q_0, Z) = 1, \\
& \varphi(q_1, b, Z, q_0, Z) = 1, \quad \varphi(q_1, \$, \tau, q_1, \tau) = 1, \quad \varphi(q_1, a, 1, q_3, 12) = 1, \\
& \varphi(q_1, b, 1, q_0, 1) = 1, \quad \varphi(q_1, a, 2, q_0, 2) = 1, \quad \varphi(q_1, b, 2, q_3, 21) = 1, \\
& \varphi(q_2, a, Z, q_3, Z2) = 1, \quad \varphi(q_2, b, Z, q_3, Z1) = 1, \quad \varphi(q_2, \$, Z, q_0, Z) = 1, \\
& \varphi(q_2, a, 1, q_2, \varepsilon) = 1, \quad \varphi(q_2, b, 1, q_0, 12) = 1, \quad \varphi(q_2, \$, 1, q_0, 1) = 1, \\
& \varphi(q_2, a, 2, q_0, 21) = 1, \quad \varphi(q_2, b, 2, q_2, \varepsilon) = 1, \quad \varphi(q_2, \$, 2, q_0, 2) = 1, \\
& \forall \sigma \in \{a, b, \$\} \quad \varphi(q_3, \sigma, Z, q_3, Z) = 1, \\
& \varphi(q_3, a, 1, q_3, 1) = 1, \quad \varphi(q_3, b, 1, q_3, 11) = 1, \quad \varphi(q_3, \$, 1, q_2, 1) = 1, \\
& \varphi(q_3, a, 2, q_3, 22) = 1, \quad \varphi(q_3, b, 2, q_3, 2) = 1, \quad \varphi(q_3, \$, 2, q_2, 2) = 1,
\end{aligned}$$

other values of arguments yield 0. □

Let us consider language which is not recognizable by any deterministic pushdown automaton:

Theorem 3.2. *Language $L_3 = \{\omega \in (a, b, c)^* \mid |\omega|_a = |\omega|_b = |\omega|_c\}$ is recognizable by a QPA with probability $\frac{2}{3}$.*

Proof. Sketch of proof. The automaton takes three equiprobable actions, during the first action it compares $|\omega|_a$ to $|\omega|_b$, whereas during the second action $|\omega|_b$ to $|\omega|_c$ is compared. Input word is rejected if the third action is chosen. Acceptance probability totals $\frac{2}{3}$. □

Theorem 3.3. *Language $L_5 = \{\omega \in (a, b, c)^* \mid |\omega|_a = |\omega|_b \text{ xor } |\omega|_a = |\omega|_c\}$ is recognizable by a QPA with probability $\frac{4}{7}$.*

Proof. Sketch of proof. The automaton starts the following actions with the following amplitudes:

- a) with an amplitude $\sqrt{\frac{2}{7}}$ compares $|\omega|_a$ to $|\omega|_b$.
- b) with an amplitude $-\sqrt{\frac{2}{7}}$ compares $|\omega|_a$ to $|\omega|_c$.
- c) with an amplitude $\sqrt{\frac{3}{7}}$ accepts the input. If exactly one comparison gives positive answer, input is accepted with probability $\frac{4}{7}$. If both comparisons gives positive answer, amplitudes, which are chosen to be opposite, annihilate and the input is accepted with probability $\frac{3}{7}$. □

Language L_5 cannot be recognized by deterministic pushdown automata.

An open problem is to find a language, not recognizable by probabilistic pushdown automata as well.

References

- ABFGK 99. A. Ambainis, R. Bonner, R. Freivalds, M. Golovkins, M. Karpinski: Quantum finite multitape automata. *Lecture Notes in Computer Science*, 1999, Vol. 1725, pp. 340-348.
- ABFK 99. A. Ambainis, R. Bonner, R. Freivalds, A. Kikusts: Probabilities to Accept Languages by Quantum Finite Automata. *Lecture Notes in Computer Science*, 1999, Vol. 1627, pp. 174-183.
- AF 98. A. Ambainis, R. Freivalds: 1-Way Quantum Finite Automata: Strengths, Weaknesses and Generalizations. *Proc. 39th FOCS*, 1998, pp. 332-341.
- BV 97. E. Bernstein and U. Vazirani: Quantum complexity theory. *SIAM Journal on Computing*, 26:1411-1473, 1997.
- BP 99. A. Brodsky, N. Pippenger: Characterizations of 1-Way Quantum Finite Automata. <http://xxx.lanl.gov/abs/quant-ph/9903014>
- CRSS 98. A. R. Calderbank, E. M. Rains, P. W. Shor, N. J. A. Sloane: Quantum error correction via codes over GF(4). *IEEE Transactions on Information Theory*, 1998, vol. 44, p. 1369-1387.

- De 85. D. Deutsch: Quantum Theory, the Church-Turing principle and the universal quantum computer. *Proc. Royal Society London, A400*, 1985. p. 96-117.
- DS 96. C. Dürr, M. Santha: A Decision Procedure for Unitary Linear Quantum Cellular Automata. *Proc. 37th FOCS*, 1996, pp. 38-45.
- Fe 82. R. Feynman: Simulating physics with computers. *International Journal of Theoretical Physics*, 1982, vol. 21, No 6/7, p. 467-488.
- Gr 99. J. Gruska: Quantum Challenges. *Lecture Notes in Computer Science*, 1999, Vol. 1725, pp. 1-28.
- Gu 89. E. Gurari: An introduction to the theory of computation. *Computer Science Press*, 1989.
- KLMT 99. E. Knill, R. Laflamme, R. Martinez, C.-H. Tseng: A cat-state benchmark on a seven bit quantum computer. <http://xxx.lanl.gov/abs/quant-ph/9908051>
- KW 97. A. Kondacs, J. Watrous: On the power of quantum finite state automata. In *Proc. 38th FOCS*, 1997, p. 66-75.
- Kr 99. M. Kravtsev: Quantum finite one-counter automata. *Lecture Notes in Computer Science*, 1999, Vol. 1725, pp. 431-440.
- MC 97. C. Moore, J. P. Crutchfield: Quantum automata and quantum grammars. <http://xxx.lanl.gov/abs/quant-ph/9707031>
- Sh 94. P. W. Shor: Algorithms for quantum computation: discrete logarithms and factoring. *Proc. 35th FOCS*, 1994, p. 124-134.
- Va 00. M. Valdat: The Class of Languages Recognizable by 1-Way Quantum Finite Automata is not Closed Under Union. *Proc. Quantum Computation and Learning. International Workshop*, 2000, pp. 52-64. E-print: <http://xxx.lanl.gov/abs/quant-ph/0001005>